
Microsoft®
PDC 2000
Professional Developers Conference
the defining point

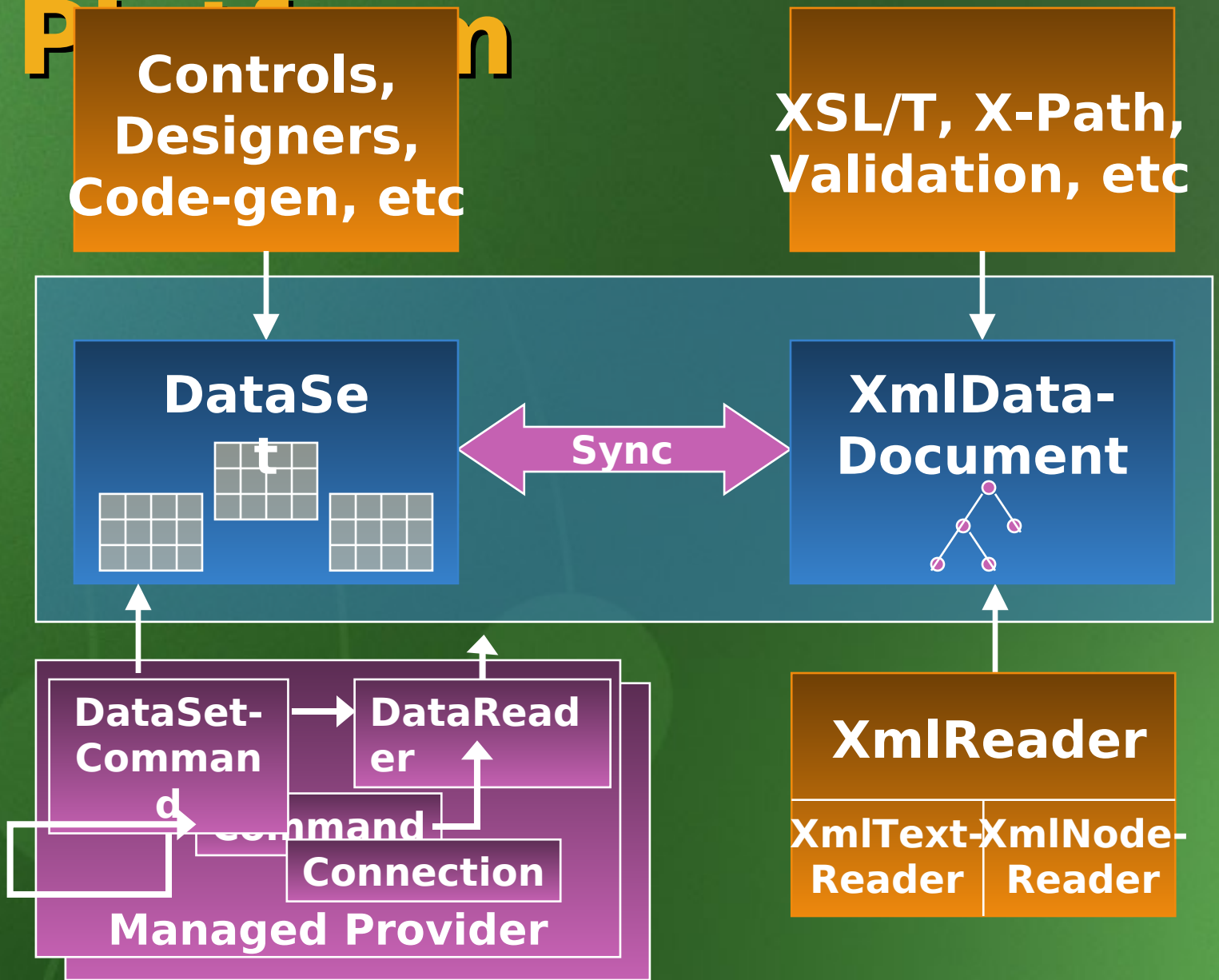
Working with Data in ASP+

Michael Pizzo
Senior Program Manager
WebData Team

Agenda

- Data in the Web Platform
- Types of Data in your Application
- How you use Data
- Summary

Data in the Web Platform



Types of Data in your Application

- Application Data
  - Generated by the application/user
    - *i.e., Shopping Cart*
- Database Data
  - Information the user queries
    - *i.e., Catalog information*
  - Persistent data that gets inserted
    - *i.e., Placed order*
- XML Data
  - Standard format for exchanging data
    - Between businesses
    - Client reach

How you use Data

- Serial Result Processing
- Frequently Used Results
- Application Data
- Singleton Query
- Web Services
- XML Data
- Database Transactions

Serial Result Processing

- Forward only processing of results
  - Generating an HTML table
  - Calculating totals
  - Binding a WebForm control
- Execute a Command and obtain a DataReader
  - Process directly

```
while(myDataReader.Read()) {
    Response.Write("Name = "+myDataReader["name"]);
}
```

  - Bind to a WebForm control

```
myList.DataSource = myDataReader();
myList.DataBind();
```

Obtaining a list of Categories

```
void UpdateCategories() {

  // Get a list of categories from the Database
  SQLConnection conn = new SQLConnection(connStr);

  conn.Open();

  // Execute a Command to get a DataReader
  SQLCommand com = new SQLCommand(
    "Select CategoryName from Categories", conn);
  SQLDataReader sqlDataReader = com.Execute();

  // Bind CategoryList to results of the DataReader
  CategoryList.DataSource = sqlDataReader;
  CategoryList.DataBind();

  // Close the Connection
  conn.Close();
}
```

Frequently Used Results

- Common Query Results
  - Fairly static data
  - Commonly used
  - Reasonable size
- Place the data in an ASP+ Cache
  - Use a DataSet as a convenient object to store results:

```
DataSet myDataSet = (DataSet)Cache["myData"];
if(null==myDataSet) {
    myDataSet = doGenerateDataSet();
    Cache["myData"] = myDataSet;
}
// use myDataSet…
```

Storing Products

```
public void CategoryList_Select(Object sender, EventArgs e) {

    DataSet productsDataSet;
    string categoryName =
       CategoryList.Items[CategoryList.SelectedIndex].Text;
     DataSet productsDataSet = (DataSet)Cache[categoryName];
     if(null==productsDataSet) {   // Generate dataset
            // Set up the Connection and DataSetCommand
            SQLConnection conn = new SQLConnection(connStr);
            SQLDataSetCommand dSCom = new SQLDataSetCommand(
              "Call GetProducts("+categoryName+")", conn);

            // Populate the DataSet and set it into the cache
            productsDataSet = new DataSet();
            dSCom.FillDataSet(productsDataSet, "products");
            Cache[categoryName] = productsDataSet;
    }

    // Bind ProductListing control to table in DataSet
    ProductListing.DataSource =
     productsDataSet.Tables[0].DefaultView;
    ProductListing.DataBind();
}
```

Application Data

- Data generated by the application
  - Application State
  - User Data
- Create a DataSet to store tabular user data
  - Store in Session State

```
DataSet userData = Session["UserTables"];
if (userData == null) {
                userData = new DataSet();
                Session["UserTables"] = userData;
}
```

Shopping Cart

```
void UpdateShoppingCart() {
  DataTable cart = (DataTable)Session["ShoppingCart"];
  if (null==cart) {  // Create a new cart
          DataSet customer = new DataSet();
    cart = customer.Tables.Add("ShoppingCart");
    cart.PrimaryKey = DataTable.CreateColumnArray(
    cart.Columns.Add("ProductID", typeof(int32)));
    cart.Columns.Add("Quantity", typeof(Int32));
    cart.Columns.Add("Name", typeof(String));
    cart.Columns.Add("Price", typeof(Double));
    cart.Columns.Add("Total", typeof(Double),
          "Price * Quantity");
    Session["ShoppingCart"] = cart;
    }

  // Bind to ShoppingCart
  ShoppingCartData.DataSource = cart.DefaultView;
  ShoppingCartData.DataBind();
}
```

Singleton Query

- Obtain a single record of information
  - Single or multiple columns
- Execute a Command and retrieve results from a DataRecord
  - ADO-Like accessors for convenience

```
customer.FirstName = myDataRecord["fname"];
```

  - Use strongly-typed accessors for best performance

```
customer.LastName = myDataRecord.GetString(1);
customer.AccountBalance = myDataRecord.GetDouble(2);
```

Product Details

```
public void ProductListing_Select(Object sender, EventArgs e) {
  int productID =
     Int32.Parse(ProdList.DataKeys[ProdList.SelectedIndex].ToString());
  DataTable shoppingCart = (DataTable)Session["ShoppingCart"];
  try {      // Row already exists
     DataRow item = shoppingCart.Rows.Find(productID);
     item["Quantity"] = (int)item["Quantity"] + 1;
  }
  catch {    // Row doesn't exist; Get info from database
     SQLConnection conn = new SQLConnection(connStr);
     conn.Open();
     SQLCommand comm = new SQLCommand(
            "Select * from Products where productID=" +
     productID,conn);
     SQLDataReader productRecord = comm.Execute();
     productRecord.Read();
     DataRow newItem = shoppingCart.NewRow();
     newItem["ProductID"]=productID;
     newItem["Quantity"]=1;
     newItem["Name"]=productRecord["ProductName"];
     newItem["Price"]=Currency.ToDouble(
            (Currency)productRecord["UnitPrice"]);
     shoppingCart.Rows.Add(newItem);
     conn.Close();
```

Passing Data to/from a WebService

- Use a DataSet
  - WebMethods know how to serialize to/from XML
  - DataSet transferred as XML in SOAP payload

```
[WebMethod]
public double ProcessOrder(DataSet ds){
        double amount = 0;
        foreach(DataRow row in ds.Tables["Orders"])
                amount += row["total"];
        return amount;
}
```

Submiting an Order

```
public void Order_Click(Object sender, ImageClickEventArgs e) {
  DataTable shoppingCart = ((DataTable)Session["ShoppingCart"]);
  OrderProcessor warehouse = new OrderProcessor();
  double orderAmount=
    warehouse.SubmitOrder(shoppingCart.DataSet);

  if(orderAmount < 0)
    Message.Text = "Invalid Order";
  else
    Message.Text = "Order Amount = " +
          double.Format(orderAmount,"$0.00");

  shoppingCart.Clear();
  UpdateShoppingCart();
}
```

Processing a Request

- Apply XSL/T transforms to XML data used in Business to Business
  - If data received as DataSet, obtain an XML View
  - i.e., BizTalk transformations
- Validate XML Document
  - According to DTD, XDR, or XSD
  - i.e., BizTalk schema

XML Transform

```
private XmlReader ProcessOrder(DataSet ds) {

    // Get an XML view of the data
    XmlDataDocument xmlData = new XmlDataDocument(ds);
    DataDocumentNavigator xmlNavigator =
            new DataDocumentNavigator(xmlData);

    // Apply a Transform
    XslTransform xsltransform = new XslTransform();
    xsltransform.Load("product.xsl";);
    return xsltransform.Transform(xmlNavigator, null);
}
```

Database Transaction

- Database Administration
  - Updating Inventory
  - Adding new information
- Inserting information into a Database
  - Logging an order
  - Debit/Credit an Account
- Control transactions of multiple statements
- Use stored procedures where available

Writing to a Database

```
private void PlaceOrder(XmlReader order) {
  // Load XML into a document
  XmlDocument xmlOrder = new XmlDocument();
  xmlOrder.Load(order);

  // Create a Command to insert the order
  SQLConnection sqlConn = new SQLConnection(ConnStr);
  sqlConn.Open();
  SQLCommand sqlCommand = new SQLCommand(
    "Insert into Orders(OrderInfo)Values(@xmlInfo)", sqlConn);

  // Add a parameter for the order
  sqlCommand.Parameters.Add(
    new SQLParameter("@XmlInfo", typeof(String), 1024));
  sqlCommand.Parameters["@xmlInfo"].Value = xmlOrder.OuterXml;

  // Execute the Command
  sqlCommand.Execute();
  sqlConn.Close();
}
```

Summary

- Microsoft provides a comprehensive, integrated set of components for working with data
  - Database Commands
    - Transactions
    - DataReader for streamed access
  - DataSet
    - Application data
    - Caching results, remoting data
  - ASP+ Cache, Session APIs
    - Performance, State management
  - XML
    - Transforms, Validation
    - Business to Business interchange
    - Client reach

Related Sessions

- Other ADO+ and XML Framework talks
  - ADO and XML Overview – Omri Gazitt
  - Using ADO+ – Mike Pizzo/Tom Kaiser
  - Data and XML in VS7 – Sean Draine
  - Using the XML Framework – Chris Lovett
  - XML in Action – William Adams
- Other WebData talks

Where do you want to go today?

Microsoft